

ΣΧΕΔΙΑΣΜΌΣ ΣΥΣΤΗΜΆΤΩΝ VLSI

5^η Εργαστηριακή Άσκηση

Γιώργος Ντάκος 1059569

7 Ιουνίου 2021

Περίληψη

Στο παρακάτω κείμενο παραδίδεται η αναφορά της 5^{ης} εργαστηριακής άσκησης του μαθήματος Σχεδιασμός Συστημάτων VLSI . Όλοι οι κώδικες έχουν συγγραφεί σε γλώσσα **VHDL & C++** μέσω του **Notepad++ & HLS** και έχουν μεταγλωττιστεί και επιβεβαιωθεί οι λειτουργίες τους μέσω του εργαλείου **ModelSim HLS**. Τέλος η σύνθεση όλων των κυκλωμάτων πραγματοποιήθηκαν από το εργαλείο **VIVADO & HLS** της **Xilinx**.

Περιεχόμενα

Ενότητα A	4
Ενότητα B	6

Κατάλογος Σχημάτων

1	Αποτελεσματα simulation	4
---	-----------------------------------	---

Κατάλογος Πινάκων

1	Αποτελέσματα Σύνθεσης Vivado	4
2	Αποτελέσματα Σύνθεσης Vivado HLS	5
3	Αποτελέσματα Σύνθεσης Vivado HLS - B	6

Ενότητα A

A.1

Υλοποιήθηκε element-wise πολλαπλασιασμός δύο πινάκων 3×3 :

$$C[i][j] = A[i][j] \cdot B[i][j], \quad i, j \in \{0, 1, 2\},$$

με στοιχεία εισόδου **unsigned** 5-bit (0–31). Το αποτέλεσμα κάθε γινομένου απαιτεί **10-bit** εύρος, καθώς $31 \cdot 31 = 961$. Ο VHDL κώδικας χρησιμοποιεί το `numeric_std` και μετατροπές σε **unsigned** για την πράξη του πολλαπλασιασμού. Μετά την προσομοίωση (testbench) πραγματοποιήθηκε σύνθεση στο Vivado και καταγράφηκαν οι πόροι και η εκτίμηση χρονισμού.

Matrix	Row 0	Row 1	Row 2
MATRIX_A	(00 00 00) (00 00 00) (00 00 00)	(00 00 00) (00 00 00) (00 00 00)	(00 00 00) (00 00 00) (00 00 00)
MATRIX_B	(00 00 00) (00 00 00) (00 00 00)	(00 00 00) (00 00 00) (00 00 00)	(00 00 00) (00 00 00) (00 00 00)
MATRIX_C	(00 00 00) (00 00 00) (00 00 00)	(00 00 00) (00 00 00) (00 00 00)	(00 00 00) (00 00 00) (00 00 00)

Σχήμα 1: Αποτελεσματα simulation

Circuit	LUTs	IO	FlipFlops	BRAM	Critical Path (ns)	Clock (MHz)	Fmax(MHz)
5BitsMatrixMultiplier3X3	198(0.25%)	180(72.00%)	0	0	4.844	0	206.440

Table 1: Αποτελέσματα Σύνθεσης Vivado

A.2

Η ίδια λειτουργία υλοποιήθηκε σε C++ στο Vivado HLS, χωρίς χρήση pragmas. Εκτελέστηκαν διαδοχικά C Simulation, C Synthesis και C/RTL Co-Simulation. Το Vivado HLS παρήγαγε RTL με επιπλέον **λογική ελέγχου** (handshake) και **θύρες μνήμης** τύπου BRAM για τους πίνακες (σήματα `ap_start/ap_done`, `ap_clk/ap_rst`, καθώς και `address/ce/q` για A,B και `address/we/d` για C). Αυτό διαφέρει από το χειροποίητο VHDL της A1, το οποίο περιγράφει άμεσα την συνδυαστική πράξη element-wise χωρίς controller ή memory interface.

A.3

Στην έκδοση A3 προστέθηκαν pragmas με στόχο την ελαχιστοποίηση του latency. Η βασική ιδέα είναι η αύξηση του παραλληλισμού: με `ARRAY_PARTITION` οι πίνακες σπάνε σε ανεξάρτητα στοιχεία/τράπεζες ώστε να είναι δυνατές ταυτόχρονες προσπελάσεις, ενώ με `UNROLL` οι επαναλήψεις των βρόχων εκτελούνται παράλληλα. Έτσι ο HLS συνθέτει πιο παράλληλο datapath, μειώνοντας σημαντικά το κρίσιμο μονοπάτι και (τυπικά) τους κύκλους latency, με αντίτιμο αύξηση πόρων (π.χ. LUTs/επιπλέον λογική interface).

Σε σχέση με την A2, η παραγόμενη VHDL της A3 περιλαμβάνει επιπλέον δομές ελέγχου/διασύνδεσης ώστε να υποστηριχθεί ο αυξημένος παραλληλισμός και οι προσπελάσεις μνήμης.

Ισχύει	ΛΥΤς	ΦαίεΦλοπς	BRAM	ΔΣΠ	Λοσκ (MHz)	ρίτσιαλ Πατη (ns)	Φμαξ(MHz)	Λατενςψ(ςψςζες)	Λατενςψ(αβςολυτε)(ns)
ματρίξ_μυλ_μυν_A2	128(0.00%)	81(0.00%)	0	0	100	6.212	160.978	25	250
ματρίξ_μυλ_μυν_A3	153(0.00%)	0(0.00%)	0	0	100	3.080	324.675	0	0

Πίνακας 2: Αποτελέσματα Σύνθεσης Vivado HLS

Στην A3 το Vivado HLS αναφέρει latency 0 cycles. Αυτό αντιστοιχεί σε πλήρως συνδυαστική υλοποίηση (combinational datapath), όπου το αποτέλεσμα παράγεται στον ίδιο κύκλο (χωρίς ενδιάμεσους καταχωρητές), λόγω του πλήρους παραλληλισμού (unrolling/partitioning). Σε αυτή την περίπτωση το κόστος μεταφέρεται στο κρίσιμο μονοπάτι, το οποίο καθορίζει την εκτιμώμενη μέγιστη συχνότητα λειτουργίας.

Ενότητα B

B.1

Στο Part B υλοποιείται κανονικός πολλαπλασιασμός πινάκων:

$$C = A \times B$$

όπου:

$$C[i][j] = \sum_{k=0}^2 A[i][k] \cdot B[k][j].$$

Τα στοιχεία των πινάκων A και B είναι unsigned 12-bit (0–4095). Κάθε στοιχείο του πίνακα C είναι άθροισμα τριών γινομένων 12-bit. Κάθε γινόμενο απαιτεί 24 bits και οι δύο προσθέσεις προσθέτουν περίπου δύο επιπλέον bits, επομένως απαιτούνται τουλάχιστον 26 bits για αποφυγή overflow.

B.2

Στην πρώτη υλοποίηση (baseline) δεν χρησιμοποιήθηκαν pragmas. Το Vivado HLS παράγει σειριακό datapath με επαναχρησιμοποίηση των πόρων, με αποτέλεσμα μεγάλο latency αλλά μικρό αριθμό DSP. Η υλοποίηση ολοκληρώνεται σε 79 κύκλους ρολογιού, δηλαδή 790 ns για 100 MHz.

Στη δεύτερη υλοποίηση εφαρμόστηκαν pragmas όπως UNROLL και ARRAY_PARTITION, ώστε να αυξηθεί ο παραλληλισμός. Όλοι οι πολλαπλασιασμοί και προσθέσεις εκτελούνται ταυτόχρονα, με αποτέλεσμα το latency να μειωθεί σε 1 κύκλο (10 ns).

Το τίμημα είναι η σημαντική αύξηση των πόρων (ιδίως DSPs), καθώς και η αύξηση του κρίσιμου μονοπατιού, επειδή σε έναν μόνο κύκλο εκτελείται πολύ μεγαλύτερη συνδυαστική λογική.

Παρατηρείται ότι παρότι η έκδοση με pragmas έχει μικρότερο latency, το κρίσιμο μονοπάτι αυξάνεται (7.52 ns αντί 6.56 ns). Αυτό οφείλεται στο ότι περισσότερες πράξεις εκτελούνται στον ίδιο κύκλο, με αποτέλεσμα μεγαλύτερο συνδυαστικό βάθος και περισσότερη λογική ελέγχου. Πρόκειται για κλασικό trade-off μεταξύ latency και χρονισμού.

Το Vivado HLS παράγαγε RTL (VHDL) με σήματα ελέγχου (ap_clk, ap_start, ap_done) και θύρες μνήμης τύπου BRAM για τους πίνακες, σε αντίθεση με το χειροποίητο VHDL της Ενότητας A, το οποίο ήταν καθαρά συνδυαστικό.

Circuit	LUTs	FlipFlops	BRAM	DSP	Clock (MHz)	Critical Path (ns)	Fmax(MHz)	Latency(cycles)	Latency(absolute)(ns)
matrix_mul_B_nopragmas	184(0.00%)	54(0.00%)	0	1(0.00%)	100	6.556	152.532	79	790
matrix_mul_B_pragmas	15(0.00%)	218(0.00%)	0	27(6.00%)	100	7.520	132.978	1	10

Table 3: Αποτελέσματα Σύνθεσης Vivado HLS - B